B. Amendments to the Claims:

1-4. (Withdrawn)

5. (Currently Amended) A method used in computer-aided circuit design for comparing two finite-precision arithmetic circuits for equivalence, each finite-precision arithmetic circuit being represented by a data flow graph, the two finite-precision arithmetic circuits being compared for equivalence by checking the equivalence of the corresponding data flow graphs, ~~data flow graphs for equivalence, each data flow graph representing a finite-precision arithmetic circuit~~, ~~the~~ each of the two data flow graphs of each of the two finite-precision arithmetic circuits comprising a plurality of edges wherein each edge has a finite bit width, an edge being further connected to a source port and one or more sink ports, the source port and the sink ports linked to an edge further connecting the edge to a plurality of arithmetic operators, each port further having a required precision and an information content associated with it, the required precision of a port being the maximum bits of the port observed at any of the output of the port, the information content of a port being the minimum bits required to represent the output of an arithmetic operation, the method comprising the steps of:

   a. determining ~~the~~ edges of each of the two data flow graphs, having information loss, wherein an edge has an information loss if the information content of the corresponding source port and the required precision of any of the corresponding sink ports are greater than the bit-width of the edge;

   b. canonizing each of the two data flow graphs, the each of the two data flow graphs being canonized by canonizing the edges having information loss, the edges being canonized by pushing the arithmetic operations that generate information loss to the end of the data flow graph ~~circuit~~;

   c. ~~dividing~~ splitting each of the two data flow graphs at the edges having information loss, into lossless sub-graphs, ~~the data flow graphs being split into lossless subgraph~~, each of the lossless sub-graphs representing an infinite-

precision circuit, the edges having information loss acting as output signals of the lossless sub-graphs;

d. balancing each of the lossless sub-graphs of each of the two data flow graphs, the each of the lossless sub-graphs being balanced to minimize the information losses at the edges of each of the lossless sub-graphs; ~~refining the information content of the ports in the data flow graphs, the information content being refined by applying the Huffman Principle and the principles of associativity of addition and multiplication, and the distributivity of multiplication over addition~~;

~~e. verifying the presence of information loss, the information loss being verified at edges corresponding to the ports for which the refinement of information content is performed;~~

~~f.~~ e. leveling each of the lossless sub-graphs of each of the two data flow graphs in the topological order of each of the data flow graphs, the level being the order of existence of each of the lossless ~~subgraphs~~ sub-graphs in the corresponding data flow graph, whereby each ~~data flow graph~~ of the lossless sub-graphs is leveled for comparison of ~~subgraphs~~ two lossless sub-graphs at the same level; and

f. checking the equivalence of the two data flow graphs using the corresponding lossless sub-graphs, the step of checking the equivalence comprising the steps of:

~~g.~~i. comparing the lossless ~~subgraphs~~ sub-graphs ~~with~~ lying at level zero, in each of the two data flow graphs ~~lowest level number~~, for ~~equality~~ equivalence, wherein the lossless ~~subgraphs~~ sub-graphs lying at level zero, being compared only if the input signals to each of the two ~~the~~ data flow graphs have the same bit-width;

~~h.~~ii. comparing the bit-width of the output edges of the lossless ~~subgraphs~~ sub-graphs lying at the same level in each of the two data flow graphs, the output edges having information loss, wherein the bit-width of the output edges of

the lossless sub-graphs being compared, if the each of the preceding lossless ~~subgraphs~~ sub-graphs are compared equal;

~~i.~~iii. comparing the lossless ~~subgraphs~~ sub-graphs with next higher level number in each of the two data flow graphs for equivalence, wherein the lossless ~~subgraphs~~ sub-graphs with next higher level number being compared only, ~~comparison of next set of being performed~~ if each of the preceding lossless ~~subgraphs~~ sub-graphs are compared equal, and the bit width of the preceding edges are same; and

repeating the steps ~~h and i,~~ ii and iii, ~~till~~ until ~~a~~ lossless sub-graph being used for comparison is the last lossless sub-graph in the corresponding data flow graph; ~~output of the circuits is reached~~.

6. (Currently Amended) The method as recited in claim 5 further comprising the steps of:

a. computing the required precision of the source port and the sink ports corresponding to each edge wherein the required precision is computed in the reverse topological order of ~~the~~ a finite-precision arithmetic circuit; and

b. determining the information content of the source port and the sink ports corresponding to each edge, wherein the information content being computed in the topological order of the finite-precision arithmetic circuit.

7-9. (Cancelled)

10. (Currently amended) The method as recited in claim 5, wherein the step of comparing the lossless ~~subgraphs~~ sub-graphs for ~~equality~~ equivalence comprises the steps of:

a. generating expressions for the lossless sub-graphs that are being compared, ~~the two data flow graphs~~ each of the lossless sub-graphs representing an infinite-

precision arithmetic circuit, the expressions being generated not having finite edge widths; and

b. comparing the lossless sub-graphs by checking the equality of the corresponding generated expressions, the equality of the generated expressions being checked, by using a theorem proving technique.

11. (Currently amended) The method as recited in claim 10, wherein the theorem proving technique is ~~step of checking the equality of the expressions comprises the step of testing the equality using~~ a Cooperating Validity Checker (CVC) lite program.

12-15. (Withdrawn)

16. (Currently Amended) A computer program product, the computer program product comprising a computer usable medium having a computer readable program code embodied therein for comparing two finite-precision arithmetic circuits for equivalence, each finite-precision arithmetic circuit being represented by a data flow graph, the two finite-precision arithmetic circuits being compared for equivalence by checking the equivalence of the corresponding data flow graphs, ~~data flow graphs for equivalence, each data flow graph representing a finite-precision arithmetic circuit~~, ~~the~~ each of the two data flow graphs comprising a plurality of edges wherein each edge has a finite bit width, an edge being further connected to a source port and one or more sink ports, the source port and the sink ports linked to an edge further connecting the edge to a plurality of arithmetic operators, each port further having a required precision and an information content associated with it, the required precision of a port being the maximum bits of the port observed at any of the output of the port, the information content of a port being the minimum bits required to represent the output of an arithmetic operation, the computer program code performing steps of:

a. determining ~~the~~ edges having information loss, for each of the two data flow graphs, wherein an edge has an information loss if the information content of the corresponding source port and the required precision of any of the corresponding sink ports are greater than the bit-width of the edge;

b. canonizing each of the two data flow graphs, the each of the two data flow graphs being canonized by canonizing the edges having information loss, the edges being canonized by pushing the arithmetic operations that generate information loss to the end of the data flow graph ~~circuit~~;

c. ~~dividing~~ splitting each of the two data flow graphs, at the edges having information loss, into lossless sub-graphs, ~~the data flow graphs being split into lossless subgraph~~, each of the lossless sub-graphs representing an infinite-precision circuit, the edges having information loss acting as output signals of the lossless sub-graphs;

d. balancing each of the lossless sub-graphs of each of the two data flow graphs, each of the lossless sub-graphs being balanced to minimize the information losses at the edges of each of the lossless sub-graphs; ~~refining the information content of the ports in the data flow graphs, the information content being refined by applying using the Huffman Principle and the principles of associativity of addition and multiplication, and the distributivity of multiplication over addition~~;

~~e. verifying the presence of information loss, the information loss being verified at edges corresponding to the ports for which the refinement of information content is performed;~~

~~f.~~ e. leveling each of the lossless sub-graphs of each of the two data flow graphs in a topological order of each of the data flow graphs, the level being the order of existence of each of the lossless ~~subgraphs~~ sub-graphs in the corresponding data flow graph, whereby each ~~data flow graph~~ of the lossless sub-graphs is leveled for comparison of ~~subgraphs~~ two lossless sub-graphs at the same level; and

f. checking the equivalence of the two data flow graphs using the corresponding lossless sub-graphs, the step of checking the equivalence comprising the steps of:

~~g~~.i. comparing the lossless ~~subgraphs~~ sub-graphs ~~with~~ lying at level zero, in each of the two data flow graphs ~~lowest level number~~, for ~~equality~~ equivalence, wherein the lossless ~~subgraphs~~ sub-graphs lying at level zero, being compared only if the input signals to each of the two ~~the~~ data flow graphs have the same bit-width;

~~h~~.ii. comparing the bit-width of the output edges of the lossless ~~subgraphs~~ sub-graphs, lying at the same level, in each of the two data flow graphs, the output edges having information loss, wherein the bit-width of the output edges of the lossless sub-graphs being compared if the each of the preceding lossless ~~subgraphs~~ sub-graphs are compared equal;

~~i~~.iii. comparing the lossless ~~subgraphs~~ sub-graphs with next higher level number in each of the two data flow graphs for equivalence, wherein the lossless ~~subgraphs~~ sub-graphs with next higher level number being compared only, ~~comparison of next set of being performed~~ if each of the preceding lossless ~~subgraphs~~ sub-graphs are compared equal and the bit width of the preceding edges are same; and

repeating the steps ~~h and i,~~ ii and iii, ~~till~~ until a lossless sub-graph being used for comparison is the last sub-graph in the corresponding data flow graph; ~~output of the circuits is reached~~.

17. (Currently Amended) The computer program product described in claim 16, wherein the computer program code further performs the steps of:

a. computing the required precision of the source port and the sink ports corresponding to each edge wherein the required precision is computed in the reverse topological order of ~~the~~ a finite-precision arithmetic circuit; and

b. determining the information content of the source port and the sink ports corresponding to each edge, wherein the information content being computed in the topological order of the finite-precision arithmetic circuit.

18. (New) The method as recited in claim 5, wherein each of the lossless sub-graphs are balanced by using the Huffman Principle and the principles of associativity of addition and multiplication, and the distributivity of multiplication over addition.

19. (New) The method as recited in claim 5, wherein the leveling each of the lossless sub-graphs of each of the two data flow graphs in the topological order comprises the steps of:

a. assigning a level number zero to a lossless sub-graph of the data flow graph, if the lossless sub-graph does not have at least one predecessor lossless sub-graph in the corresponding data flow graph; and

b. assigning a level number as one plus the maximum of the level numbers of all the predecessors of a lossless sub-graph, if the lossless sub-graph has at least one predecessor lossless sub-graph in the corresponding data flow graph.